CLAIMS

1. A method to protect a data file against loss of data, wherein the data file comprises a set of N data blocks that are stored in N respective nodes, comprising:

generating a k x n matrix of code blocks;

storing K code blocks in K respective nodes, distinct from the N respective nodes, wherein each code block has an i^{th} code block computed as: $C_i = f(g_i 1(A_1), \ldots g_i n(A_n))$, where *g* is a permutation operator that comprises a superposition of cyclic permutations.

2. The method as described in claim 1 wherein the superposition of cyclic permutations is of the form: $b_0 * c^0 + b_1 * c^1 + \ldots b_k c^k + b_{(m-1)} * c^{(m-1)}$, where b_k is a bit (0 or 1), c^0 is an identity, and c^k is a cycle operation c repeated k times.

3. The method as described in claim 2 wherein the K code words comprise code words associated with the following permutation operators:

[1 2 1 4]

[1 3 3 5].

4. The method as described in claim 2 wherein the K code words comprise code words associated with the following permutation operators:

[1 1 1 1 1 1]

[1 2 3 4 5 6]

[1 4 5 16 17 20].

5. The method as described in claim 2 wherein the K code words comprise code words associated with the following permutation operators:

[2 2 2 1 1 3]
[2 6 3 3 2 2]
[1 3 5 6 3 1].

6. The method as described in claim 1 wherein the function f() is a bitwise exclusive OR (XOR) operation.

7. The method as described in claim 1 wherein the nodes comprise a heterogeneous redundant array of independent nodes (RAIN).

8. A method of storing data comprising a set of N data blocks, comprising the unordered steps of:

generating a k x n matrix of code blocks;

storing the N data blocks in N respective nodes; and

storing K code blocks in K respective nodes, distinct from the N respective nodes, wherein each code block has an i^{th} code block computed as: $C_i = f\,(g_i 1(\, A_1\,), \ldots g_i\, n\,(A_n))$, where g is a permutation operator that comprises a superposition of cyclic permutations.

9. The method as described in claim 8 wherein the superposition of cyclic permutations is of the form: $b_0 * c0 + b_1 * c1 + \ldots b_k c^k + b_{(m-1)} * c^{(m-1)}$, where b_k is a bit (0 or 1), c^0 is an identity, and c^k is a given cycle operation c repeated k times.

10. The method as described in claim 8 wherein the function f( ) is a bitwise exclusive OR (XOR) operation.

11. In a redundant array of independent nodes, wherein a data file comprising a set of N data blocks are stored in N respective nodes of the array, a method of protecting the data file against loss of data, comprising:

storing K code blocks in K respective nodes of the array, distinct from the N respective nodes, wherein each code block has an i^{th} code block computed as: $C_i = f(g_i1(A_1), \ldots g_i n(A_n))$, where *g* is a permutation operator that comprises a superposition of cyclic permutations.

12. The method as described in claim 11 further including the step of recovering a portion of the data file using the K code blocks.

13. The method as described in claim 12 wherein the step of recovering a portion of the data file performs a matrix inversion on a diagonal sub-matrix derived from the K code blocks.

14. The method as described in claim wherein the step of recovering a portion of the data file includes the steps of:

performing a given operation on an available portion of a K code block using a key to recover an additional portion of the K code block; and

repeating the above step until the K code block is recovered sufficiently to enable the portion of the data file to be recovered.

15. The method as described in claim 11 wherein the superposition of cyclic permutations is of the form: $b_0 * c^0 + b_1 * c^1 + \ldots b_k c^k + b_{(m-1)} * c^{(m-1)}$, where b_k is a bit (0 or 1), c^0 is an identity, and c^k is a given cycle operation c repeated k times.

16. A process for protecting against loss and for enhancing accessibility in storage or memory, or protecting against loss and enhancing speed of transmission on communication paths, of information that is represented in storage or memory, or represented as data signals on communication paths, the information comprising N data blocks, comprising:

dispersing the information by transmitting the N data blocks in the form of said data signals carried on multiple first communication paths, or by storing the N data blocks in N storage or memory locations;

dispersing protection information by transmitting K code blocks in the form of data signals carried on multiple second communication paths distinct from the multiple first communications paths, or by storing the K code blocks in K storage or memory locations distinct from the N storage or memory locations, wherein each code block has an i^{th} code block computed as: $C_i = f(g_i 1(A_1), \ldots g_i n(A_n))$, where g is a permutation operator that comprises a superposition of cyclic permutations.

17. The process as described in claim 16 further including the step of recovering a portion of the information using the K code blocks.

18. The process as described in claim 17 wherein the step of recovering a portion of the information performs a matrix inversion on a diagonal sub-matrix derived from the K code blocks.

19. The process as described in claim 17 wherein the step of recovering a portion of the information includes the steps of:

performing a given operation on an available portion of a K code block using a key to recover an additional portion of the K code block; and

repeating the above step until the K code block is recovered sufficiently to enable the portion of the information to be recovered.

20. The process as described in claim 16 wherein the superposition of cyclic permutations is of the form: $b_0 * c^0 + b_1 * c^1 + \ldots b_k c^k + b_{(m-1)} * c^{(m-1)}$, where b_k is a bit (0 or 1), c^0 is an identity, and c^k is a given cycle operation c repeated k times.